Nigerian Journal of Technology (NIJOTECH)
Vol. 32, No. 1, March, 2013, pp. 37–??.

University of Nigeria. ISSN 1115-8443

DYNAMIC-THRESHOLD-LIMITED TIMED-TOKEN (DTLTT) PROTOCOL

C. Kalu[1,a], **S. Ozuomba**[2,a], **G.N. Onoh**[b]

[a]Department of Electrical/Electronic & Computer Engineering, University of Uyo, Akwa Ibom State, Nigeria.
Emails: [1]*conskay4@yahoo.com;* [2]*simeonoz@yahoo.com*
[b]Dept. of Electrical/Electronic Engineering, Enugu State University of Science and Technology Owerri, Nigeria.

Abstract

An improved version of the Static-Threshold-Limited On-Demand Guaranteed Service Timed-Token (STOGSTT) Media Access Control (MAC) protocol for channel capacity allocation to the asynchronous traffic in Multiservice Local Area Network (MLANs) was developed and analyzed. STLODGSTT protocol uses static value of threshold bandwidth to allocate available bandwidth to the asynchronous traffic, as such, the throughput of STLODGSTT protocol drops significantly under non-uniform heavy load of asynchronous traffic. The DTLTT protocol dynamically adjusts the threshold bandwidth in response to the variations in the load distribution of the asynchronous traffic. In view of this dynamic mechanism, under various load distributions of the asynchronous traffic, the DTLTT protocol maintains higher throughput than the STLODGSTT protocol. The improvement is demonstrated through analytical computations and simulation results.

Keywords: multi-access, multiservice, network, synchronous, asynchronous, traffic, timed-token

1. Introduction

Nowadays, efficient support for both time-critical and best-effort traffic in the same Local Area Networks (LANs) is essential [1, 2] The MAC protocols for such MLANs must provide not only bounded message transmission time, as required by the hard and soft real-time tasks, but also high throughput, as demanded by non real-time tasks that relies on best-effort services [3, 4, 5, 6]. An attractive MAC approach for such networks is the timed-token protocol. Consequently, the timed-token protocol has been incorporated into several high-bandwidth network standards [7], such as, IEEE802.4 Token Bus LAN [8] , Fiber Distributed Data Interface (FDDI) [9, 10, 11, 12, 13] SAFENET [14], Manufacturing Automation Protocol (MAP) [15], High-Speed Ring Bus [16], in PROFIBUS [17], and in wireless networks [18, 19, 20].

The STOGSTT protocol is a version of the timed-token protocol developed to improve the communication services provided by the existing timed-token protocols [21]. Nevertheless, the throughput of the STOGSTT protocol drops whenever some of the nodes do not have asynchronous traffic to transmit. That means, if n is the number of nodes that are heavily loaded with asynchronous traffic, where $1 \leq n \leq N$, the throughput of the STOGSTT protocol drops whenever $n < N$, but it attains to its maximum throughput when $n = N$. The problem is solved in the DTLTT protocol developed in this paper. The DTLTT protocol dynamically adjust the value of the threshold bandwidth used in allocating available bandwidth to the n heavily loaded nodes. With this dynamic mechanism, the DTLTT maintains higher throughput irrespective of the variations in the distribution of asynchronous traffic in the network.

The rest of the paper is organized as follows; the network model , message model and timed token protocol parameters are presented in Section 2 along with a review of the STOGSTT protocol. The DTLTT protocol is presented and analyzed in Section 3. In section 4, the simulation and analytical computation results are presented and discussed. Finally, concluding remarks and recommendations for further studies are stated in Section 5.

2. Network Characteristics

2.1. Network Model

2.1.1. Token ring network model

The study considered a token ring or logical ring network consisting of N nodes (or stations), as shown in Fig 1. Each node has a unique number in the range

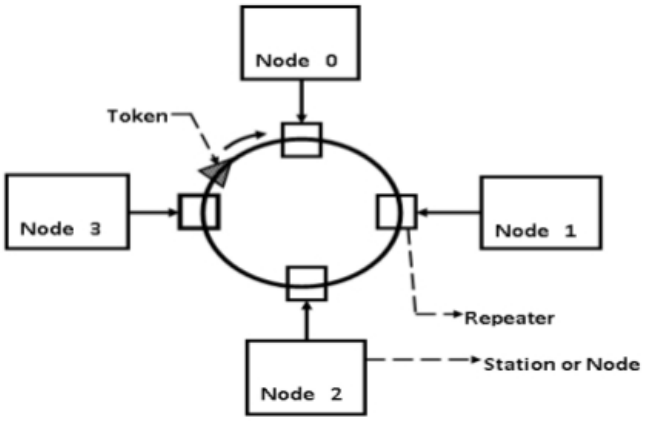


Figure 1: A 4-station token ring network.

0, 1, 2, ..., $N-1$. In addition, each node is connected to two other neighbouring nodes by unidirectional point-to-point media that form a single closed path. For each node i, the next node along the unidirectional medium is station $(i+1)$ or more appropriately node $(i+1)$ mod N.

2.1.2. Logical ring network model

Similar network model can also exist on a logical ring network otherwise known as token bus network as, as shown in Fig 2. In this case, we consider a bus with N nodes connected to form a unidirectional logical ring on the bus. Each node has a unique number in the range 0, 1, 2, ..., $N-1$. In addition, each node has two other neighbouring nodes. For each node i, the next node along the unidirectional logical ring is station $(i+1)$ or more appropriately node $(i+1)$ mod N.

2.1.3. The token and ring latency

A special bit pattern called the token circulates around the ring (logical ring or token ring) from node i to nodes $i+1$, $i+2$, ... until node $i+(N-1)$, then to nodes i, $i+1$, $i+2$, ..., helping to determine which node should send a frame of message among the contending nodes.

Definition: Let w_i denote the latency or walk-time between a node i and its upstream neighbor node $(i+1)$. w_i can also be defined as the time needed to transmit the token between nodes, including the overhead introduced by the protocol [2]. Then,

$$W = \sum_{i}^{N-1} w_i \tag{1}$$

The ring latency, W denotes the token walk time around the ring when none of the nodes in the network disturb it [9, 22].

2.2. Message Model

Messages generated in the system at run time may be classified as either synchronous messages or asynchronous messages. In the following discussion it is assumed that there is one stream of synchronous messages on each node [23, 9]. It is also assumed that the network is free from hardware or software failures. Hence, in the N-node ring, the synchronous message set, M, consist of N streams of synchronous messages; s_0, s_1, s_2, ..., s_{N-1} where, $M = \{s_0, s_1, s_2, \ldots, s_{N-1}\}$. The synchronous message stream, s_i at node i is given as $s_i = \{P_i, C_i, D_i\}$ and shown in Fig 3 where:

- **Message length,** C_i is the maximum amount of time required to transmit a stream message. This includes the time required to transmit both the payload data and message headers.
- **Period Length,** P_i is the minimum inter-arrival period between consecutive messages in stream, s_i at node i. If the first message of node i is put in the transmission queue at time $t_{i,1}$, then the j-th message in stream s_i will arrive at time $t_{i,j} = t_{i,1} + (j-1)P_i$, where $j > 1$. For instance, if the first message arrives at time t, then the second message will arrive at $t+P_i$ and the third message will arrive at $t+2P_i$ as shown in Fig 2.
- **Message Deadline,** D_i, is the relative deadline associated with messages in stream s_i, that is, the maximum amount of time that can elapse between a message arrival and the completion of its transmission. Thus, the transmission of the j-th message in stream s_i that arrives at $t_{i,j}$ must be completed no later than $t_{i,j} + D_i$, which is the message's absolute deadline. Again, as an example, if the first message in the message stream, s_i arrives at time t, then it must be transmitted not later than $t + D_i$, as shown in Fig 2.

2.3. The timed-token medium access control (MAC) protocol parameters

The parameters of the timed-token as presented in [21] include the following:

a) *Target Token Rotation Time, (TTRT)*: TTRT is the time needed by the token to complete an entire round-trip of the network. The value of TTRT is denoted as τ.

b) *Synchronous Capacity of Node i* (H_i): H_i represents the maximum time for which a station, i is allowed to transmit synchronous messages during every token receipt. Then according to [21],

$$H_0 + H_1 + \ldots + H_{N-1} = \left(\sum_{i=0}^{i=N-1} (H_i) \right) = H \tag{2}$$

$$H_i + w_i = \sigma_i \tag{3}$$

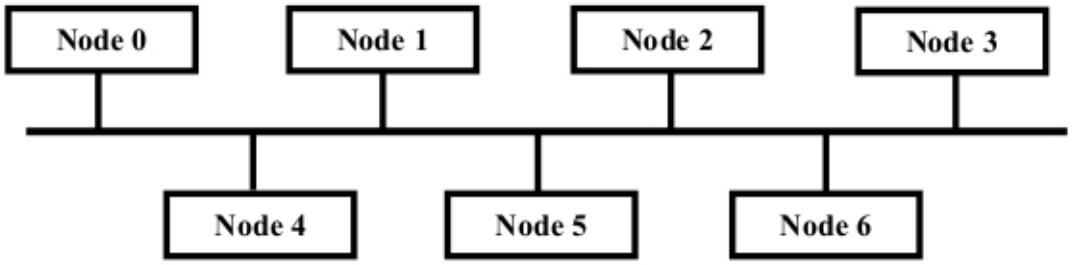


Figure 2: A 7-station token bus network.

$$H + W = T \tag{4}$$

$$A = \tau + T \tag{5}$$

where A be defined as the total time units available to the asynchronous traffic in every cycle.

Constraints:

For proper operation of the timed-token protocol, the choice of values for the τ and T parameters must satisfy the *Protocol Constraint* and the *Deadline Constraint* which according to [21] is given as,

$$\tau \leq \min_{i=0,1,\ldots,N-1}(D_i) \tag{6}$$

c) *Token Rotation Timer of Node i* (TRT_i): TRT_i is the cycle length or the time between two consecutive token receipts at node i.

d) *The Unused Synchronous Bandwidth*, (ε): According to [21]

$$h_i = H_i - \varepsilon_i \tag{7}$$

where h_i denotes the used portion of H_i and ε_i the unused portion of H_i time units reserved for the synchronous traffic in node i (where $h_i \leq H_i$). Then, for a system that is lightly loaded with synchronous traffic, out of the H_i time units, only h_i time units are used in node i leaving ε_i time units unused

$$\varepsilon_i + \varepsilon_i + \ldots + \varepsilon_i = \varepsilon \tag{8}$$

e) *An Asynchronous-Limit Variable of Node i* (THT_i): THT_i is used to control the amount of time for which node i can transmit asynchronous messages [21].

f) *The Number of Heavily Loaded Nodes in a Network* (n): n denotes the number of nodes that are heavily loaded with asynchronous traffic out of the N nodes in the network. In uniformly heavily loaded network $n = N$, however, in a non-uniformly heavily loaded network $1 \leq n < N$

2.4. Review of STLODGSTT protocol

Generally, in the timed token protocols, there is unallocated bandwidth per cycle, given as $A = \tau - T$ (which is the same as, $A = \tau - H - W$). The STLODGSTT allocates this A bandwidth to the asynchronous traffic using its static threshold limited best-effort bandwidth allocation mechanism. Specifically, the STLODGSTT protocol uses a_i time units in each node i, for the transmission of asynchronous traffic, where $a_i = max(0, \tau - \varepsilon - (t_i - t_{i-N})) + A_T$, and $A_T = \frac{\tau - T}{N}$ [21]. According to [21], in each cycle, the total time units used for the transmission of asynchronous traffic is given as $\sum_{1=0}^{i=N-1}(a_i) \leq A$ time units (where $A = \tau - H - W$). At its stable state, STLODGSTT allocates A_T time units to each node for its asynchronous traffic. Hence, at its best, the STLODGSTT can have an average of $n*(\frac{\tau - T}{N})$ time units per cycle for its asynchronous traffic; where n is the number of nodes with heavy load of asynchronous traffic. When $n < N$, the throughput of STLODGSTT per cycle for the asynchronous traffic drops. In order to overcome this constraint on the STLODGSTT, the DTLTT protocol is developed in this paper to dynamically determine n and hence recalculates A_T with respect to n rather than N. Thus, in the DTLTT protocol, $A_T = \frac{\tau - T}{n}$. In this case, $n * (\frac{\tau - T}{N})$ will always remain at its maximum value of$\tau - T$.

3. The DTLTT MAC Protocol

The flowchart of the DTLTT protocol is presented in Fig 4 while the detailed algorithm is given here as *Protocol Q MAC Algorithm*. The algorithm is adapted from that of the STOGSTT Protocol [21]. At each token receipt the DTLTT algorithm determines n, the number of nodes with heavy load of asynchronous traffic and hence recalculates $A_T = \frac{\tau - T}{n}$. The updated value of A_T is then used in the allocation of available bandwidth to the asynchronous traffic among the n nodes that are heavily loaded with asynchronous traffic in that given cycle.

3.1. Outline of the DTLTT MAC Algorithm: Protocol Q

Q1: Initialization Cycle:

Q1.1 Define TTRT (that is τ) and N

Q1.2 Define w_i for $i = 0, 1, \ldots N - 1$.

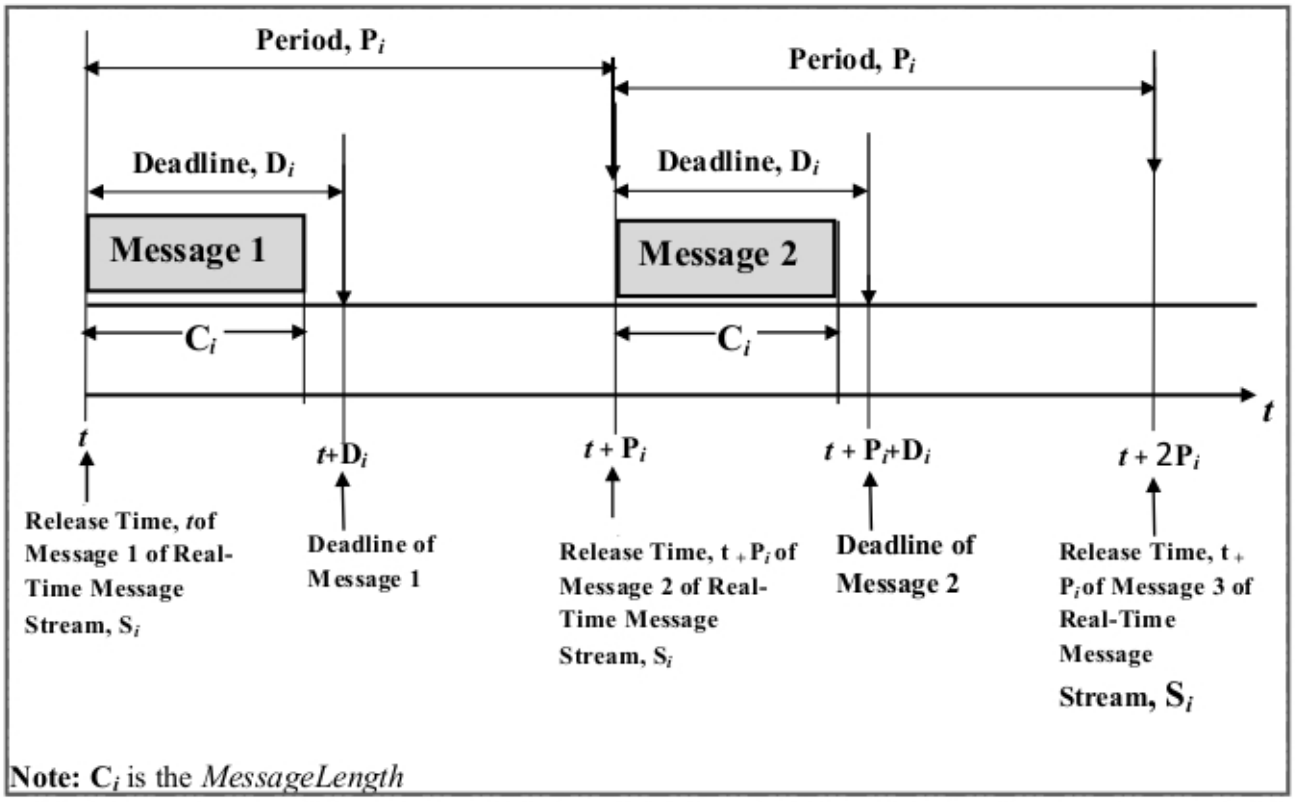
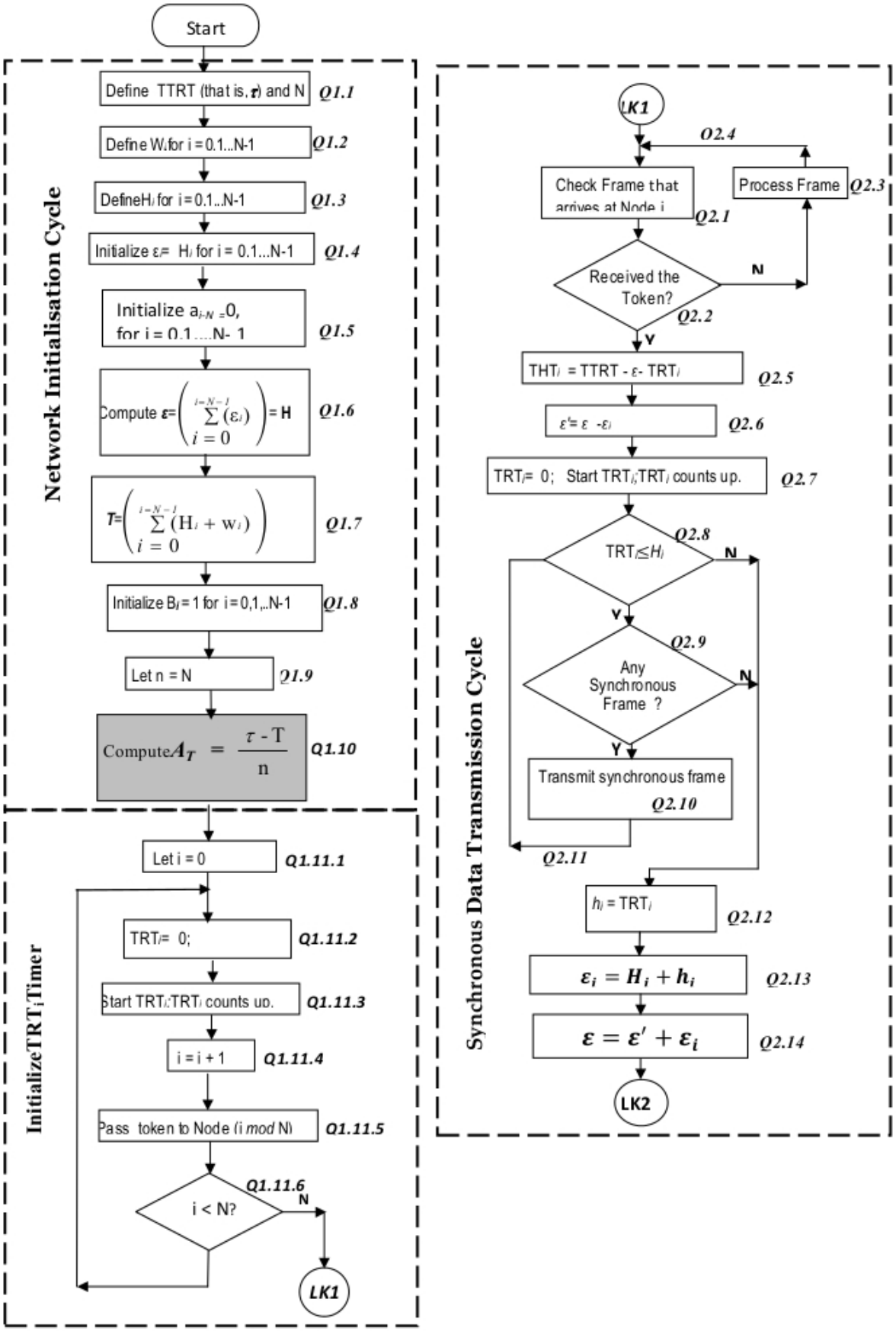


Figure 3: Model for the synchronous (or real-time) message stream, S_i in node 1 [21].

Q1.3 Define H_i for $i = 0, 1, \ldots N-1$.
Q1.4 Initialize $\varepsilon_{\mathbf{i}} = H_i$ and $\mathbf{h_i} = 0$ for $i = 0, 1, \ldots N-1$.
Q1.5 Initialize $a_{i-N} = 0$ for $i = 0, 1, \ldots N-1$.
Q1.6 Compute $\varepsilon = \sum_{i=0}^{i=N-1} (\varepsilon_i) = H$.
Q1.7 Compute $T = \left(\sum_{i=0}^{i=N-1} (H_i + w_i) \right)$;
Q1.8 Initialize $B_i = 1$ for $i = 0, 1, \ldots N-1$.
Q1.9 $n = N$.
Q1.10 Compute $A_T = \frac{\tau - T}{n}$.
Q1.11 Initialize Token rotation timer (TRT_i) Timer. **Q1.11.1** $i = 0$. **Q1.11.2** $TRT_i = 0$. **Q1.11.3** Start TRT_i; TRT_i Counts up. **Q1.11.4** $i = i + 1$. **Q1.11.5** Pass the Token to Node $i + 1$. **Q1.11.6** IF ($i < N$) Then Goto Step **Q1.11.2** Else Goto Step **Q2.1** End if.

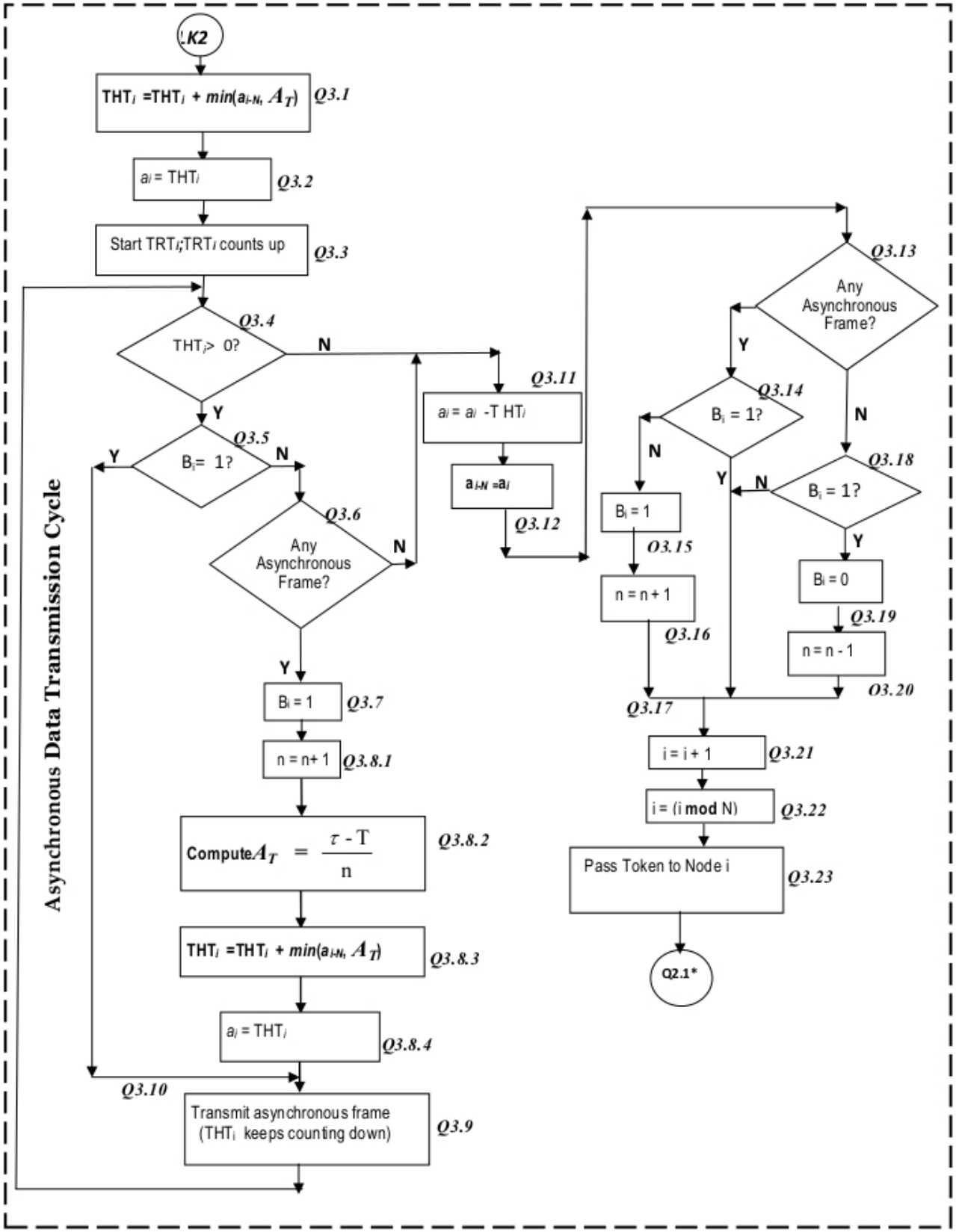
Q2: Data Transmission Cycle,
I: Transmission Of Synchronous Frames
Q2.1 Check Frames that arrives at Node i.
Q2.2 IF (Frames is Token) Then Goto Step **Q2.5** Else Goto Step **Q2.3** End if.
Q2.3 Process Frame (Store, Ignore, etc.)
Q2.4 Goto Step **Q2.1**.
Q2.5 $THT_i = \max(TTRT - \varepsilon - TRT_i, 0)$.
Q2.6 $\varepsilon' = \varepsilon - \varepsilon_I$.
Q2.7.1 $TRT_i = 0$. **Q2.7.2** Start TRT_i **Q2.7.3** TRT_i counts up.
Q2.8 IF ($TRT_i \leq H_i$) Then Goto Step **Q2.9** Else Goto Step **Q2.12** End if.
Q2.9 IF (Synchronous Data Available) Then Goto Step **Q2.10** Else Goto Step **Q2.11** End if.
Q2.10 Transmit Synchronous Frames.
Q2.11 Goto Step **Q2.8**.
Q2.12 $h_i = TRT_i$.
Q2.13 $\varepsilon_i = H_i + h_i$.
Q2.14 $\varepsilon = \varepsilon' + \varepsilon_I$ (Goto **Q3.1**).

Q3: Data Transmission Cycle,
Part i: Transmission Of Asynchronous Frames
Q3.1 $THT_i = THT_i + \min(a_{1-N}, A_T)$.
Q3.2 $a_i = THT_i$.
Q3.3 Start THT_i, THT_i counts down.
Q3.4 IF ($THT_i > 0$) Then Goto Step **Q3.5** Else Goto Step **Q3.11** End if.
Q3.5 IF ($B_i = 1$) Then Goto Step **Q3.9** Else Goto Step **Q3.6** End if.
Q3.6 IF (Asynchronous Data Available) Then Goto Step **Q3.7** Else Goto Step **Q3.11** End if.
Q3.7 $B_i = 1$.
Q3.8.1 $n = n + 1$. **Q3.8.2** Compute $A_T = \frac{\tau - T}{n}$. **Q3.8.3** $THT_i = THT_i + \min(a_{i-N}, A_T)$. **Q3.8.4** $a_i = THT_i$.
Q3.9 Transmit Asynchronous Frame.



(a) Continues on next page.

Figure 4: Flowchart of DTLTT protocol.



(b) Continued from previous page.

Figure 4: Flowchart of DTLTT protocol.

Q3.10 Goto Step **Q3.4**.
Q3.11 $a_i = a_i - THT_i$.
Q3.12 $a_{i-N} = a_i$.
Q3.13 IF (Asynchronous Data Available) Then Goto Step **Q3.14** Else Goto Step **Q3.18** Endif.
Q3.14 IF ($B_i = 1$) Then Goto Step **Q3.15** Else Goto Step **Q3.17** End if.
Q3.15 $B_i = 1$.
Q3.16 $n = n + 1$.
Q3.17 Goto Step **Q3.21**.
Q3.18 IF ($B_i = 1$) Then Goto Step **Q3.19** Else Goto Step **Q3.21** End if.
Q3.19 $B_i = 0$.
Q3.20 $n = n - 1$.
Q3.21 $i = i + 1$.
Q3.22 $i = (i \mod N)$.
Q3.23 Pass the Token to Node i (Goto **Q2.1**).

3.2. Analysis of the DTLTT MAC algorithm

According to the protocol operations in **Q2.5** of Protocol Q MAC Algorithm in Section 3.1, when the token arrives at node i, then THT_i is determined as,

$$THT_i \leq \max(0,\, TTRT - \varepsilon - TRT_i) \quad \text{for all } TTRT_i \tag{9}$$

According to the protocol operations in **Q3.1** and **Q3.8.3** of Protocol Q MAC Algorithm in Section 3.1, THT_i is updated as follows

$$THT_i = THT_i + \min(A_T,\, a_{i-N}) \tag{10}$$

According to the protocol operations in **Q3.2** and **Q3.8.4**, a_i is defined as; $a_i = THT_i$; thus, from Eq 13a **Which 13a?????**

$$a_i = THT_i + \min(A_T,\, a_{i-N}) \tag{11}$$

Since only n nodes are heavily loaded, where $n \leq N$, some nodes may have no asynchronous frames to transmit; in that case,

$$a_i = 0 \tag{12}$$

Let a_x represent those nodes that are heavily loaded with asynchronous traffic, where, $x \exists n$; $x = 0, 1, 2, \ldots, n-1$ and $1 \leq n \leq N$. Let a_y represent those nodes that are not heavily loaded with asynchronous traffic $y \nexists n$ and $y = 0, 1, 2, \ldots, n'$. Where

$$n' = N - n \tag{13}$$

Now $A = \tau - T$ and $A_T = \frac{A}{n}$; where n is the number of nodes that are heavily loaded with asynchronous traffic and $1 \leq n \leq N$. For a system that is heavily loaded with asynchronous traffic, in every cycle, at least n nodes have as much asynchronous traffic as A_T, such that when each of the n nodes transmits A_T asynchronous frames in the cycle, a total of A asynchronous frames would have been transmitted in every cycle.

Further, for those nodes that are heavily loaded with asynchronous traffic,

$$a_x \leq A_T;\ \text{where } i \exists n$$

Thus, for those nodes that are heavily loaded with asynchronous traffic, $a_{i-N} = a_x$, then

$$\min(A_T,\, a_{i-N}) = A_T;\ \text{where } i \exists n$$

Hence, from Eq13b **Which 13b?????**

$$a_i \leq THT_i + A_T;\ \ where i \exists n \tag{14}$$

Similarly, for those nodes that are not heavily loaded with asynchronous traffic, $ay = 0$ and $a_{i-N} = a_y$, then

$$a_{i-N} = 0;\ \text{where } i \nexists n$$

Thus, for those nodes that are not heavily loaded with asynchronous traffic,

$$\min(A_T,\, a_{i-N}) = 0;\ \text{where } i \nexists n$$

Hence, for those nodes that are not heavily loaded with asynchronous traffic

$$a_i = 0;\ \ where i \nexists n \tag{15}$$

For a system that is heavily loaded with asynchronous traffic, at least one node is heavily loaded with asynchronous traffic, hence, Eq13, Eq14 and Eq15,

$$a_i \leq \max(0,\, TTRT - \varepsilon - TRT_i) + A_T \tag{16}$$

If TRT_i is replaced with $t_i - t_{i-N}$, and $TTRT$ with τ then, Eq16 gives;

$$ai \leq \tau - \varepsilon - (t_i - t_{i-N}) + A_T \text{ when } \tau - \varepsilon > (t_i - t_{i-N})$$

or

$$a_i = A_T\ when \tau - \varepsilon = (t_i - t_{i-N}) \tag{17}$$

Again, for a heavily loaded system, these give;

$$a_i \leq \max(0,\, \tau - \epsilon - (t_i - t_{i-N})) + A_T \text{ for all } i > 0 \tag{18}$$

If the token reaches node i at time, t_i, then the token will reach node $i+1$ at t_{i+1} after transmitting h_i time units of synchronous traffic and a_i time units of asynchronous traffic along with a token walk-time, w_i. Thus

$$t_{i+1} \leq t_i + a_i + h_i + w_i \tag{19}$$

$$t_{i+1} \leq t_{i-N} + A_T + \tau - \varepsilon + (\sigma_i - \varepsilon_i) \text{ for } (t_i - t_{i-N}) < \tau - \epsilon$$

$$t_{i+1} \leq t_i + A_T + \sigma_i - \varepsilon_i \text{ for } \tau - \varepsilon = (t_i - t_{i-N}) < \tau - \epsilon \tag{20}$$

Table 1a: Mean(a_i) for the STOGSTT and DTLTT protocols; where $H = 80$, $\varepsilon = 40$ and $n = 1, 2, 3, 4$.

Cycle K	$n=1$		$n=2$		$n=3$		$n=4$	
	STLODGSTT	DTLTT	STLODGSTT	DTLTT	STLODGSTT	DTLTT	STLODGSTT	DTLTT
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00
2	10.00	16.00	10.67	10.67	10.67	10.67	10.67	10.67
3	10.00	16.00	13.33	16.00	12.00	12.00	12.00	12.00
4	10.00	16.00	13.33	16.00	15.00	16.00	12.80	12.80
5	10.00	16.00	13.33	16.00	15.00	16.00	16.00	16.00
6	10.00	16.00	13.33	16.00	15.00	16.00	16.00	16.00
7	10.00	16.00	13.33	16.00	15.00	16.00	16.00	16.00
8	10.00	16.00	13.33	16.00	15.00	16.00	16.00	16.00

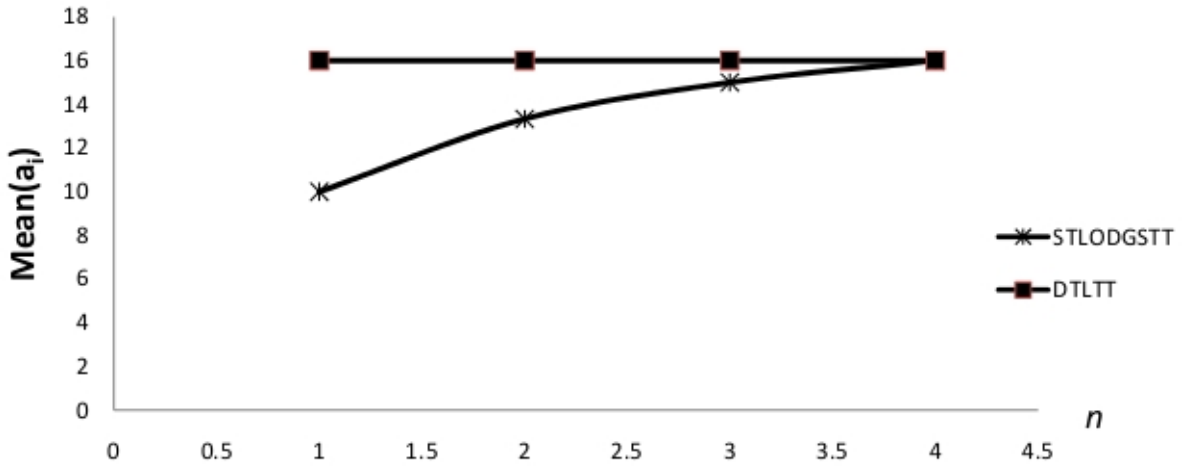


Figure 5: Variation of Mean(a_i) with n for the STOGSTT/DTLTT protocols, where $H = 80$ and $\varepsilon = 40$; Data from Table 1, Cycle $K = 10$.

Table 1b: Steady state Mean(ai) for the STOGSTT and DTLTT protocols extracted from Table 1a ; for all cycle, $K \geq 5$.

n	Mean(a_i) for STOGSTT	Mean(a_i) for DTLTT
1	10	16
2	13.33	16
3	15	16
4	16	16

Then, combining these for a system that is heavily loaded gives

$$t_{i+1} \leq \max(t_i + A_T,\ t_{i-N} + A_T + \tau - \varepsilon) + (\sigma_i - \varepsilon_i) \text{ for all } i \geq 0 \tag{21}$$

where $\sigma_i - \varepsilon_i = \sigma_{(i \mod N)} - \varepsilon_{(imodN)}$.

Eq19 for t_{i+1} in the DTLTT protocol is the same for t_{i+1} in the STOGSTT Protocol [21]. As such the remaining set of steps for the analysis in [21] apply to the DTLTT protocol. The only difference between the remaining analytical expressions in the DTLTT protocol and that of STOGSTT Protocol in [21] is that whereas $A_T = \lfloor \frac{\tau - T}{n} \rfloor$, in the DTLTT protocol; $A_T = \lfloor \frac{\tau - T}{N} \rfloor$ in the STOGSTT Protocol. Thus, following the analysis in [21] , from Eq 21,

$$t_i \leq \left(1 + \lfloor \frac{i}{N+1} \rfloor\right)(\tau - T) + \lfloor \frac{i}{N+1} \rfloor A_T + \lfloor \frac{i-1}{N+1} \rfloor (T - \varepsilon) + \left(\sum_{j=0}^{j=(i-1) \mod N} (\sigma_j - \varepsilon_j) \right) \tag{22}$$

and

$$t_{Nk} \leq \left(1 + \lfloor \frac{i}{N+1} \rfloor\right)(\tau - T) + \lfloor \frac{n \cdot K}{N+1} \rfloor A_T + k(T - \varepsilon) \tag{23}$$

3.2.1. Upper Bound On Cycle Length, $\max(t_i - t_{i-N})$

Also, following the analysis in [21],

$$\max(t_i - t_{i-N}) = \tau \text{ for all } i \geq 0 \text{ and } \varepsilon = 0 \tag{24}$$

3.2.2. Average Cycle Length, ($\hat{c}$)

Let $\hat{c}$ be the Average Cycle Length, and from protocol **Q3.8.2**

$$A_T = \lfloor \frac{\tau - T}{n} \rfloor \tag{25}$$

Table 2a: Mean(TRT_i) for STOGSTT and DTLTT protocols, where $H = 80$, $\varepsilon = 40$ and $n = 1, 2, 3, 4$.

Cycle	$n=1$		$n=2$		$n=3$		$n=4$	
K	STLODGSTT	DTLTT	STLODGSTT	DTLTT	STLODGSTT	DTLTT	STLODGSTT	DTLTT
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
2	32.00	32.00	21.33	21.33	21.33	21.33	21.33	21.33
3	54.00	60.00	41.33	41.33	31.00	31.00	31.00	31.00
4	54.00	60.00	57.33	60.00	46.00	46.00	36.80	36.80
5	54.00	60.00	57.33	60.00	59.00	60.00	48.80	48.80
6	54.00	60.00	57.33	60.00	59.00	60.00	60.00	60.00
7	54.00	60.00	57.33	60.00	59.00	60.00	60.00	60.00
8	54.00	60.00	57.33	60.00	59.00	60.00	60.00	60.00

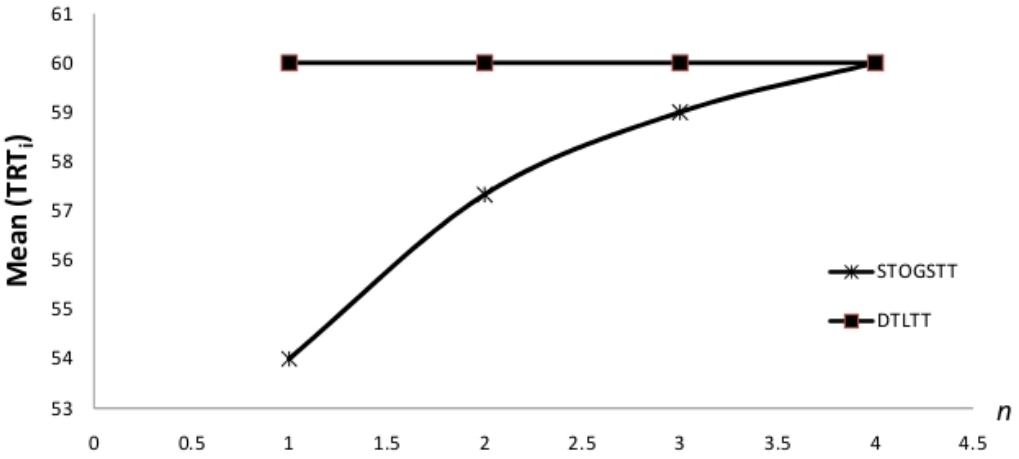


Figure 6: Variation of Mean(TRT_i) with n for the STOGSTT and DTLTT protocols, where $H = 80$ and $\varepsilon = 40$; Data from Table 2, Cycle $K = 10$.

Table 2b: Steady state Mean(TRT_i) for the STOGSTT and DTLTT protocols extracted from Table 2a for all cycle, $K \geq 5$.

n	Mean(TRT_i) for STOGSTT	Mean(TRT_i) for DTLTT
1	54	60
2	57.33	60
3	59	60
4	60	60

Then, from Eq23, $\hat{c}$ is given as

$$\hat{c} \leq lim_{k\to\infty}(\frac{t_{Nk}}{k}) \leq \lfloor\frac{n(\tau-T)}{n+1}\rfloor + \lfloor\frac{n(A_T)}{n+1}\rfloor + (T-\varepsilon)$$

$$\hat{c} \leq \lfloor\frac{n(\tau-T)}{n+1}\rfloor + \lfloor\frac{\tau-T}{n+1}\rfloor + (T-\varepsilon) \quad (26)$$

$$\hat{c} \leq (\tau-T) + (T-\varepsilon)$$

$$\hat{c} \leq \tau - \varepsilon \quad (27)$$

These give the average cycle length for the DTLTT protocol under light load of synchronous traffic (where $\varepsilon > 0$) but with non uniform heavy load of asynchronous traffic. When $\varepsilon = 0$,

$$\hat{c} \leq (\tau-T) + T$$

$$\hat{c} \leq \tau \quad (28)$$

These give the average cycle length for the DTLTT protocol under heavy load of synchronous traffic where $\varepsilon = 0$ but with non uniform heavy load of asynchronous traffic.

3.2.3. (Channel Capacity) Time Used By The Asynchronous Traffic Per Cycle, A_V

Let A_V denote the average (Channel Capacity) time used by the asynchronous traffic per cycle in the DTLTT, then from Eq26

$$A_v \leq \lfloor\frac{n(\tau-T)}{n+1}\rfloor + \lfloor\frac{n(A_T)}{n+1}\rfloor \quad (29)$$

Where from Eq25 $A_T = \lfloor\frac{(\tau-T)}{n}\rfloor$. Eq29 gives the average time used by the asynchronous traffic per cycle irrespective of the load level of the synchronous traffic.

3.3. Comparison Of the Performance of The DTLTT Protocol and The STLODGSTT Protocol Under Non Uniform Heavy Load of Asynchronous Traffic. The Average Asynchronous Traffic Time Units Per Cycle (A_V)

In this paper, it has been shown that for the DTLTT protocol, the average Asynchronous Traffic (Capacity) Time Units Per Cycle for the DTLTT under non

uniform heavy load of asynchronous traffic is given as A_{VD}, where, from Eq 26 or Eq29, $A_{VD} = A_V$. Hence,

$$A_{VD} \leq \lfloor \frac{n(\tau - T)}{n+1} \rfloor + \lfloor \frac{n(A_T)}{n+1} \rfloor$$

Where from Eq25, $A_T = \lfloor \frac{\tau - T}{n} \rfloor$, thus,

$$A_{VD} \leq \lfloor \frac{n(\tau - T)}{n+1} \rfloor + \lfloor \frac{n}{n+1} \left(\frac{\tau - T}{n} \right) \rfloor$$

$$A_{VD} \leq \tau - T \quad (30)$$

These give the average time used by the asynchronous traffic per cycle in the DTLTT irrespective of the load level of the synchronous traffic. On the other hand, it has been shown in [21], the average time used by the asynchronous traffic per cycle in the STLODGSTT protocol, is given as A_{VS}, where

$$A_{VS} \leq \lfloor \frac{N(\tau - T)}{N+1} \rfloor + \lfloor \frac{N(A_T)}{N+1} \rfloor \quad (31a)$$

With respect to Eq26a and Eq26b, under non uniform heavy load of asynchronous traffic, Eq31a gives

$$A_{VS} \leq \lfloor \frac{n(\tau - T)}{n+1} \rfloor + \lfloor \frac{n(A_T)}{n+1} \rfloor \quad (31b)$$

Now , in the STLODGSTT protocol, irrespective of the load distribution of the asynchronous traffic, $A_T = \lfloor \frac{\tau - T}{N} \rfloor$, thus, Eq31b gives

$$A_{VS} \leq \lfloor \frac{n(\tau - T)}{n+1} \rfloor + \lfloor \frac{n}{n+1} \left(\frac{\tau - T}{N} \right) \rfloor \quad (31c)$$

For any given τ, T, N and n, A_{VD} always exceeds A_{VS} where,

$$AVD - AVS = \lfloor \frac{n(\tau - T)}{n+1} \left(\frac{N - n}{nN} \right) \rfloor \quad (32)$$

3.4. Simulation of Protocol Q

The simulation of the MAC algorithms for the DTLTT protocol (Protocol Q) was conducted with a program written with Visual Basic for Applications (VBA). The program runs in Microsoft Office Excel 2007 environment. The following mathematical expressions will be used to compare the simulation results with the results obtained from the analytical computations. For the simulation results, if the values of N, n, T, τ and ε remain constant for at least M consecutive cycles where $M \gg N$, then MEAN(TRT_i) approaches $\hat{c}$ obtained from the analytical computations where, MEAN($RT^{\sharp}_{j,k}$) is given as

$$\text{MEAN}(TRT_i) = \left(\frac{1}{n+1} \right) \left(\sum_{i=x}^{i=x+n} TRT_i \right) \text{ for } x > N \quad (33)$$

Table 3: Values of $A_{VD} - A_{VS}$ based on simulation and based on the analytic expression in Eq32b.

n	Values of AVD based on analytic computations	Values of AVS based on analytic computations	From Table 1b: Mean(ai) for the DTLTT - Mean(ai) for the STOGSTT based on simulation	Values of AVD - AVS based on analytic computations of Eq 32b
1	16.0	10.0	6.0	6.0
2	16.0	13.3	2.7	2.7
3	16.0	15.0	1.0	1.0
4	16.0	16.0	0.0	0.0

Similarly,

$$\text{MEAN}(a_i) = \left(\frac{1}{n+1} \right) \left(\sum_{i=x}^{i=x+n} a_i \right) \text{ for } x > N \quad (34)$$

The average values are considered as from cycle $N+1$ and the average values are taken for every set of $n+1$ cycles.

The Maximum Cycle Length is MAX(TRT_i) for all $i \geq 0$.

4. The protocols Simulation and Analytical Computation Results

4.1. The network and protocol parameters for the test

Consider a ring network with four stations ($N = 4$). The network uses the DTLTT and STOGSTT protocols for its MAC where the timed-token parameters are given as follows: $TTRT = \tau = 100$, $w_i = 1$ and $H_i = 20$ for all the nodes. With these given parameters, then $H = 4(20) = 80$.

4.2. For the simulation

The simulation was performed for a 4-node network (that means, N = 4) . The network is non-uniformly heavily loaded with asynchronous traffic (that means, $1 \leq n \leq N$) and also has about 50% (that means, $\varepsilon = 0.5 * H$) load level of the synchronous traffic. The simulation data captured are Mean(a_i) (the Mean of the time units allocated to the asynchronous traffic in every cycle) and Mean(TRT_i) (the Mean of Token Rotation Time in every cycle). The simulation results for DTLTT and STOGSTT protocols are shown in Table 1a, Table 1b, Table 2a, Table 2b, and Table 3, as well as in Fig 5 and Fig 6.

4.3. Protocols test results

Note: In Table 3, A_{VS} is the same thing as Mean(a_i) for the STOGSTT and A_{VD} is the same thing as Mean(a_i) for the DTLTT. A_{VS} and A_{VD} are obtained from analytical computations whereas Mean(a_i) for the two protocols are obtained from the simulation of the algorithms for STOGSTT and DTLTT respectively.

4.4. Discussion of result

4.4.1. Validating the analytical results with the simulation results

From the results in Table 3, it is seen that the values of $A_{VD} - A_{VS}$ obtained from the analytical expression in Eq32b corresponds with the values of Mean(a_i) for the DTLTT - Mean(a_i) for the STOGSTT obtained from the simulation results and shown in Table 1b . Also, the value of A_{VD} obtained from the analytical expression in Eq30c as $A_{VD} = \tau - T = 16$ for all values of n corresponds with the values of A_{VD} obtained from the simulation results and shown in Table 1a and in Table 1b. These results indicate that the analytical expression effectively capture the performance of the DTLTT protocol (Protocol Q).

4.4.2. Comparison of the performance of the protocols: The average asynchronous traffic time units per cycle for the timely-token protocol, the STOGSTT protocol and the DTLTT protocol

From Table 1a, Table 1d, Table 3 and Fig 5, the Average Asynchronous Traffic Time Units Per Cycle for the STOGSTT protocol increases as n increases from 1 to N . However, the Average Asynchronous Traffic Time Units Per Cycle for STOGSTT protocol trails behind that of the DTLTT Protocol . Specifically, as n increases from 1 to N, the Average Asynchronous Traffic Time Units Per Cycle for the DTLTT Protocol remains constant at the value of $\tau - T$, which in the case of Table 1a, Table 1b and Table 3, is $\tau - T = 16$. In all, the Average Asynchronous Traffic Time Units Per Cycle for the DTLTT Protocol is not affected by n, whereas in the STOGSTT protocol, their Average Asynchronous Traffic Time Units Per Cycle decreases as n decrease from N to 1.

Also, the same argument applies to Mean(TRT_i) (that is, the Average Cycle Length) for the DTLTT and STOGSTT protocols, as can be seen from Table 2a, Table 2b, and Fig 6.

5. Conclusion and Recommendations

5.1. Conclusion

In this paper, DTLTT protocol which is an improved version of the STOGSTT protocol is presented. Through analytical approach and the use of computer simulations, the DTLTT protocol is shown to maintain higher throughput irrespective of the variations in the distribution of the asynchronous load level. It therefore effectively solved the problem which is present in the STOGSTT protocol.

5.2. Recommendations

Additional improvement in the throughput of the timed token protocols can be achieved if the asynchronous traffic is allowed to use part or all of the spare bandwidth left by the synchronous traffic without introducing token lateness problem. Further studies are required to realize this additional improvement in the timed token protocols.

References

1. Ricardo M. Survey of Real-Time Communication in CSMA-Based Networks. *Network Protocols and Algorithms*, Vol. 2, No. 1, 2010, pp 158–183.
2. White P. P. RSVP and Integrated Services in the Internet: A Tutorial. *IEEE Communications Magazine*, May 1997, pp 100–106.
3. Indumathi G.and Murugesan K. A bandwidth efficient scheduling framework for non real time applications in wireless networks. *International Journal of Distributed and Parallel systems*, Vol.1, No.1, 2010, pp 46–59.
4. Zhang S. and Burns A. *Timing Properties of the Timed Token Protocol.* Tech. Rept. (YCS 243), Dept of Computer Science, Univ. of York, May 1994.
5. Zhang S. and Burns A. *A Study of Timing Properties with the Timed Token Protocol.* Technical Report (YCS 226), Dept. of Computer Sci., Univ. of York, March 1994.
6. Regnier, P. and G. Lima. *Deterministic integration of hard and soft real-time communication over shared-ethernet.* In Proc. of Workshop of Tempo Real, Curitba, Brazil, 2006.
7. Nicholas M. and Wei Z. The timed-token protocol for real-time communications. *Computer*, Vol. 27, no. 1, 1994, pp. 35-41.
8. The Institute of Electrical and Electronic Engineers. *Token-passing bus access method and physical layer specifications.* America national Standard ANSI/IEEE std. 802.4, 1985.
9. Biao C. and Wei Z. *Properties of the Timed Token Protocol.* Department of Computer Science, Texas A&M University College Station, Technical Report 92-038 TX 77843-3112, Oct., 1992.
10. Shin K.G. and Zheng Q. FDDI-M: A scheme to double FDDI's ability of supporting synchronous traffic. *IEEE Trans. on Parallel and Distributed Systems*, Vol. 6, No. 11, 1995, pp. 1125–1131.
11. Kenneth C.S. and Marjory J.J. Cycle time properties of the FDDI token ring protocol. *IEEE Trans. Software. Eng.*, Vol. SE-13, 1987, pp. 376-385.
12. Grow R. (1982). *A timed token protocol for local area networks.* Proc. Electro82, Token Access Protocols, Paper 17/3, May 1982.
13. Chan E., Chen D., Cao J. and Lee C. H. The time Properties of the FDDI-M Medium Access Control Protocol. *The Computer Journal*, Vol. 82, No. 1, 1999, pp. 96-102.
14. Dept. of Defense US. *Survivable Adaptable Fibre-Optic Embedded Networks.* MIL-STD-2004, US Dept. of Defense, Washington D.C., 1992.

15. Mcguffin L.J., Reid L.O, and Sparks S.R. MAP/TOP in CIM Distributed Computing. *IEEE Network*, vol.2, no. 3, 1988, pp. 23–31.

16. Uhlhorn R.W. The fibre-optic high-speed data bus for a new generation of a military aircraft. *IEEE LCS*, Vol.2, No.1, 1991, pp. 36–43.

17. Tovar E., and Vasques F. *Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications.* Proc. of the 15th IFAC Workshop on Distributed Computer Control Systems, 1998.

18. Lee, D., Attias, R., Puri, A., Sengupta, R., Tripakis, S. and Varaiya P. *A wireless token ring protocol for intelligent transportation systems.* IEEE Intelligent Transportation Systems Conference Proceedings, Aug. 2001, pp. 1152-1157

19. Malpani N., Vaidya N., and Welch J. *Distributed Token circulation on mobile Ad Hoc Networks.* 21th International Conference on distributed computing systems (ICDCS 2001) PHOENIX, Arizona, USA, April 2001, pp.691-701.

20. Willig A. *Analysis of the PROFIBUS Token passing protocol over wireless links.* Proc. of IEEE Int. Symposium on Industrial Electronics (IEEE-ISIE 2002), LAquila, Italy, July 2002, pp. 56-61.

21. Ozuomba S., Chukwudebeb G.A., Obot A.B.. Static-Threshold-Limited On-Demand Guaranteed Service For Asynchronous Traffic In Timely-Token Protocol. *Nigerian Journal of Technology*, Vol. 30, No. 2, 2011, pp124 - 142.

22. Ozuomba S. and Chukwudebe G.A. An Improved Algorithm For Channel Capacity Allocation In Timer Controlled Token Passing Protocols. *International Journal Of Nigerian Computer Society*, Vol. 9, No 1, 2003, pp 116–124.

23. Agrawal G., Chen B., Zhao W., and Davari S. Guaranteeing synchronous message Deadlines with the Timed Token Access Control Protocol. *IEEE Transaction on computers*, Vol 43. No.3, 1994, pp 237-239.